UP-DETR: Unsupervised Pre-training for Object Detection with Transformers

Zhigang Dai[1,2,3*], Bolun Cai[2], Yugeng Lin[2], Junying Chen[1,3†]
[1]School of Software Engineering, South China University of Technology
[2]Tencent Wechat AI
[3]Key Laboratory of Big Data and Intelligent Robot
(South China University of Technology), Ministry of Education
zhigangdai@hotmail.com, {arlencai,lincolnlin}@tencent.com, jychense@scut.edu.cn

Abstract

*Object detection with transformers (DETR) reaches competitive performance with Faster R-CNN via a transformer encoder-decoder architecture. Inspired by the great success of pre-training transformers in natural language processing, we propose a pretext task named **random query patch detection** to Unsupervisedly Pre-train DETR (**UP-DETR**) for object detection. Specifically, we randomly crop patches from the given image and then feed them as queries to the decoder. The model is pre-trained to detect these query patches from the original image. During the pre-training, we address two critical issues: multi-task learning and multi-query localization. (1) To trade off classification and localization preferences in the pretext task, we freeze the CNN backbone and propose a patch feature reconstruction branch which is jointly optimized with patch detection. (2) To perform multi-query localization, we introduce UP-DETR from single-query patch and extend it to multi-query patches with object query shuffle and attention mask. In our experiments, UP-DETR significantly boosts the performance of DETR with faster convergence and higher average precision on object detection, one-shot detection and panoptic segmentation. Code and pre-training models: https://github.com/dddzg/up-detr.*

1. Introduction

Object detection with transformers (DETR) [5] is a recent framework that views object detection as a direct prediction problem via a transformer encoder-decoder [39]. Without hand-designed sample selection [46] and non-maximum suppression, DETR reaches a competitive performance with Faster R-CNN [34]. However, DETR comes with training and optimization challenges, which needs large-scale training data and an extreme long training schedule. As shown in Fig. 1 and Section 4.1, we find that DETR performs poorly in PASCAL VOC [13], which has insufficient training data and fewer instances than COCO [28].

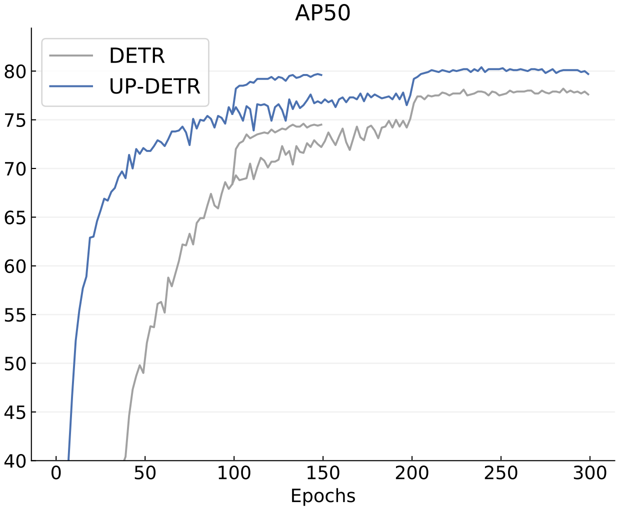


Figure 1: The VOC learning curves (AP$_{50}$) of DETR and UP-DETR with ResNet-50 backbone. Here, they are trained on `trainval07+12` and evaluated on `test2007`. We plot the short and long training schedules, and the learning rate is reduced at 100 and 200 epochs, respectively.

With well-designed pretext tasks, unsupervised pre-training models achieve remarkable progress in both natural language processing (*e.g.* GPT [32, 33] and BERT [11]) and computer vision (*e.g.* MoCo [16, 9] and SwAV [7]). In DETR, the CNN backbone (ResNet-50 [19] with ∼23.2M parameters) has been pre-trained to extract a good visual representation, but the transformer module with ∼18.0M parameters has not been pre-trained. More importantly,

[*]This work is done when Zhigang Dai was an intern at Tencent Wechat AI.

[†]Corresponding author.

although unsupervised visual representation learning (*e.g.* contrastive learning) attracts much attention in recent studies [16, 8, 14, 4, 6, 1], existing pretext tasks can not directly apply to pre-train the transformers of DETR. The main reason is that DETR mainly focuses on spatial localization learning instead of image instance-based [16, 8, 14] or cluster-based [4, 6, 1] contrastive learning.

Inspired by the great success of unsupervised pre-training in natural language processing [11], we aim to unsupervisedly pre-train the transformers of DETR on a large-scale dataset (*e.g.* ImageNet), and treat object detection as the downstream task. The motivation is intuitive, but existing pretext tasks seem to be impractical to pre-train the transformers of DETR. To overcome this problem, we propose **Unsupervised Pre-training DETR (UP-DETR)** with a novel unsupervised pretext task named **random query patch detection** to pre-train the detector without any human annotations — we *randomly* crop multiple *query patches* from the given image, and pre-train the transformers for *detection* to predict bounding boxes of these query patches in the given image. During the pre-training procedure, we address two critical issues as follows:

(1) Multi-task learning: Object detection is the coupling of object classification and localization. To avoid query patch detection destroying the classification features, we introduce **frozen pre-training backbone** and **patch feature reconstruction** to preserve the feature discrimination of transformers.

(2) Multi-query localization: Different object queries focus on different position areas and box sizes. To illustrate this property, we propose a simple single-query pre-training and extend it to a multi-query version. For multi-query patches, we design **object query shuffle** and **attention mask** to solve the assignment problems between query patches and object queries.

In our experiments, UP-DETR performs better than DETR on PASCAL VOC [13] and COCO [28] object detection with faster convergence and better average precision. Besides, UP-DETR also transfers well with state-of-the-art performance on one-shot detection and panoptic segmentation. In ablations, we find that **freezing the pre-training CNN backbone** is the most important procedure to preserve the feature discrimination during the pre-training.

2. Related Work

2.1. Object Detection

Most object detection methods mainly differ in positive and negative sample assignment. Two-stage detectors [34, 3] and a part of one-stage detectors [27, 29] construct positive and negative samples by hand-crafted multi-scale anchors with the IoU threshold and model confidence. Anchor-free one-stage detectors [38, 48, 22] assign positive and negative samples to feature maps by a grid of object centers. Zhang *et al.* [46] demonstrate that the performance gap between them is due to the selection of positive and negative training samples. DETR [5] is a recent object detection framework that is conceptually simpler without hand-crafted process by direct set prediction [37], which assigns the positive and negative samples automatically.

Apart from the positive and negative sample selection problem, the trade-off between classification and localization is also intractable for object detection. Zhang *et al.* [45] demonstrate that there is a domain misalignment between classification and localization. Wu *et al.* [40] and Song *et al.* [35] design two head structures for classification and localization. They point out that these two tasks may have opposite preferences. For our pre-training model, it maintains shared feature for classification and localization. Therefore, it is essential to take a well trade-off between these two tasks.

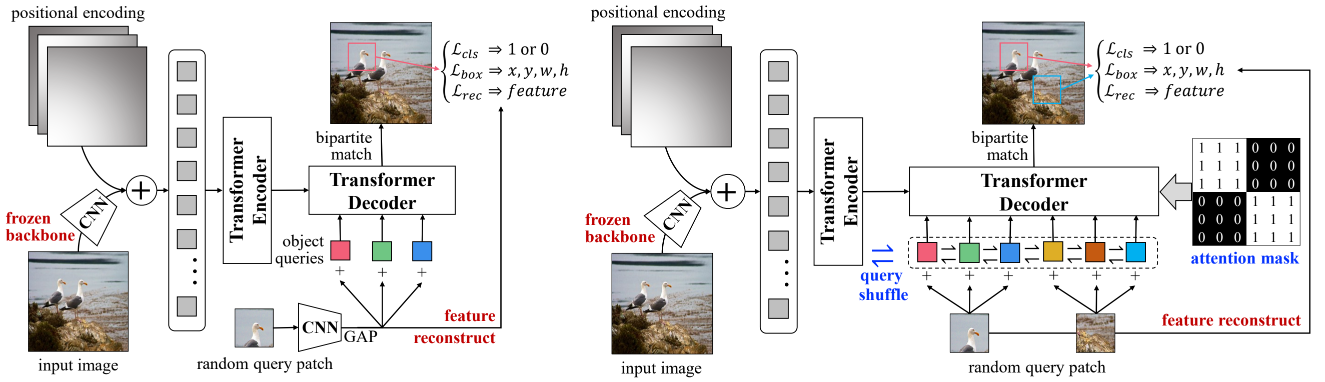
2.2. Unsupervised Pre-training

Unsupervised pre-training models always follow two steps: pre-training on a large-scale dataset with the pretext task and fine-tuning the parameters on downstream tasks. For unsupervised pre-training, the pretext task is always invented, and we are interested in the learned intermediate representation rather than the final performance of the pretext task.

To perform unsupervised pre-training, there are various of well-designed pretext tasks. For natural language processing, utilizing time sequence relationship between discrete tokens, masked language model [11], permutation language model [43] and auto regressive model [32, 33] are proposed to pre-train transformers [39] for language representation. For computer vision, unsupervised pre-training models also achieve remarkable progress recently for visual representation learning, which outperform the supervised learning counterpart in downstream tasks. Instance-based discrimination tasks [44, 41] and clustering-based tasks [6] are two typical pretext tasks in recent studies. Instance-based discrimination tasks vary mainly on maintaining different sizes of negative samples [16, 8, 14] with non-parametric contrastive learning [15]. Moreover, instance discrimination can also be performed as parametric instance classification [4]. Clustering-based tasks vary on offline [6, 1] or online clustering procedures [7]. UP-DETR is a novel pretext task, which aims to pre-train transformers based on the DETR architecture for object detection.

3. UP-DETR

The proposed UP-DETR contains pre-training and fine-tuning procedures: (a) the transformers are unsupervisedly *pre-trained* on a large-scale dataset without any human an-



(a) single-query patch (N=3, M=1)

(b) multi-query patch (N=6, M=2)

Figure 2: The pre-training procedure of UP-DETR by random query patch detection. (a) There is only a single-query patch which we add to all object queries. (b) For multi-query patches, we add each query patch to N/M object queries with object query shuffle and attention mask. CNN is not drawn in the decoder of (b) for neatness.

notations; (b) the entire model is *fine-tuned* with labeled data which is same as the original DETR [5] on the downstream tasks. In this section, we mainly describe how to pre-train the transformer encoder and decoder with random query patch detection.

As shown in Fig. 2, the main idea of random query patch detection is simple but effective. Firstly, a frozen CNN backbone is used to extract a visual representation with the feature map $f \in \mathbb{R}^{C\times H\times W}$ of an input image, where C is the channel dimension and $H \times W$ is the feature map size. Then, the feature map is added with positional encodings and passed to the multi-layer transformer encoder in DETR. For the random cropped query patch, the CNN backbone with global average pooling (GAP) extracts the patch feature $p \in \mathbb{R}^C$, which is flatten and supplemented with object queries $q \in \mathbb{R}^C$ before passing it into a transformer decoder. Noting that the *query patch* refers to the cropped patch from the original image but *object query* refers to position embeddings, which are fed to the decoder. The CNN parameters are shared in the whole model.

During the pre-training procedure, the decoder predicts the bounding boxes corresponding to the position of random query patches in the input image. Assuming that there are M query patches by random cropping, the model infers a prediction fixed-set $\hat{y} = \{\hat{y}_i\}_{i=1}^N$ corresponding to N object queries ($N > M$). For better understanding, we will describe the training details of single-query patch ($M = 1$) in Section 3.1, and extend it to multi-query patches ($M > 1$) with object query shuffle and attention mask in Section 3.2.

3.1. Single-Query Patch

DETR learns different spatial specialization for each object query [5], which indicates that different object queries focus on different position areas and box sizes. As we randomly crop the patch from the image, there is no any priors about the position areas and box sizes of the query patch. To preserve the different spatial specialization, we explicitly specify single-query patch ($M = 1$) to all object queries ($N = 3$) as shown in Fig. 2a.

During the pre-training procedure, the patch feature p is added to each different object query q, and the decoder generates N pairs of predictions $\hat{y} = \{\hat{y}_i\}_{i=1}^N$ to detect the bounding box of query patch in the input image. Following DETR [5], we compute the same match cost between the prediction $\hat{y}_{\hat{\sigma}(i)}$ and the ground-truth y_i using *Hungarian* algorithm [37], where $\hat{\sigma}(i)$ is the index of y_i computed by the optimal bipartite matching.

For the loss calculation, the predicted result $\hat{y}_i = (\hat{c}_i \in \mathbb{R}^2, \hat{b}_i \in \mathbb{R}^4, \hat{p}_i \in \mathbb{R}^C)$ consists of three elements: $\hat{c}_i$ is the binary classification of matching the query patch ($c_i = 1$) or not ($c_i = 0$) for each object query; $\hat{b}_i$ is the vector that defines the box center coordinates, its width and height $\{x, y, w, h\}$. They are re-scaled relative to the image size; $\hat{p}_i$ is the reconstructed feature with $C = 2048$ for the ResNet-50 backbone typically. With the above definitions, the *Hungarian* loss for all matched pairs is defined as:

$$\mathcal{L}(y,\hat{y}) = \sum_{i=1}^{N}[\lambda_{\{c_i\}}\mathcal{L}_{cls}(c_i,\hat{c}_{\hat{\sigma}(i)}) + \mathbb{1}_{\{c_i=1\}}\mathcal{L}_{box}(b_i,\hat{b}_{\hat{\sigma}(i)}) + \mathbb{1}_{\{c_i=1\}}\mathcal{L}_{rec}(p_i,\hat{p}_{\hat{\sigma}(i)})]. \tag{1}$$

Here, $\mathcal{L}_{cls}$ is the cross entropy loss over two classes (match the query patch $vs.$ not match), and the class balance weight $\lambda_{\{c_i=1\}} = 1$ and $\lambda_{\{c_i=0\}} = M/N$. $\mathcal{L}_{box}$ is a linear combination of ℓ_1 loss and the generalized IoU loss with the same weight hyper-parameters as DETR [5]. $\mathcal{L}_{rec}$ is the reconstruction loss proposed in this paper to balance classification and localization during the unsupervised pre-training, which will be discussed in detail below.

3.1.1 Patch Feature Reconstruction

Object detection is the coupling of object classification and localization, where these two tasks always have different feature preferences [45, 40, 35]. Different from DETR, we propose a feature reconstruction term $\mathcal{L}_{rec}$ to preserve classification feature during localization pre-training. The motivation of this term is to preserve the feature discrimination extract by CNN after passing feature to transformers. $\mathcal{L}_{rec}$ is the mean squared error between the ℓ_2-normalized patch feature extracted by the CNN backbone, which is defined as follows:

$$\mathcal{L}_{rec}(p_i, \hat{p}_{\hat{\sigma}(i)}) = \left\| \frac{p_i}{\|p_i\|_2} - \frac{\hat{p}_{\hat{\sigma}(i)}}{\|\hat{p}_{\hat{\sigma}(i)}\|_2} \right\|_2^2 . \quad (2)$$

3.1.2 Frozen Pre-training Backbone

With the patch feature reconstruction, the CNN backbone parameters seriously affect the model training. Our motivation is that the feature after transformer should have similar discrimination as the feature after the CNN backbone. Therefore, we freeze the pre-training backbone and reconstruct the patch feature after the transformers by $\mathcal{L}_{rec}$. Stable backbone parameters are beneficial to transformer pre-training, and accelerate the feature reconstruction.

As described above, we propose and apply feature reconstruction and frozen backbone to preserve feature discrimination for classification. In Section 4.5.1, we will analyze and verify the necessity of them with experiments.

3.2. Multi-Query Patches

For general object detection, there are multiple object instances in each image (*e.g.* average 7.7 object instances per image in the COCO dataset). Moreover, single-query patch may result in the convergence difficulty when the number of object queries N is large. Therefore, single-query patch pre-training is inconsistent with multi-object detection task, and is unreasonable for the typical object query setting $N = 100$. However, extending a single-query patch to multi-query patches is not straightforward, because the assignment between M query patches and N object queries is a specific negative sampling problem for multi-query patches.

To solve this problem, we divide N object queries into M groups, where each query patch is assigned to N/M object queries. The query patches are assigned to the object queries in order. For example, the first query patch is assigned to the first N/M object queries, the second query patch to the second N/M object queries, and so on. Here, we hypothesize that it needs to satisfy two requirements during the pre-training: **(1) Independence of query patches**. All the query patches are randomly cropped from the image. Therefore, they are independent without any relations. For example, the bounding box regression of the first cropping is not concerned with the second cropping. **(2) Diversity of object queries**. There is no explicit group assignment between object queries for the downstream tasks. In other words, the query patch can be added to arbitrary N/M object queries ideally.

3.2.1 Attention Mask

To satisfy the independence of query patches, we utilize an attention mask matrix to control the interactions between different object queries. The mask matrix $\mathbf{X} \in \mathbb{R}^{N \times N}$ is added to the softmax layer of self-attention in the decoder $softmax\left(QK^\top/\sqrt{d_k} + \mathbf{X}\right)\mathbf{V}$. Similar to the token mask in UniLM [12], the attention mask is defined as:

$$\mathbf{X}_{i,j} = \begin{cases} 0, & \text{i, j in the same group} \\ -\infty, & \text{otherwise} \end{cases} , \quad (3)$$

where $\mathbf{X}_{i,j}$ determines whether the object query q_i attends to the interaction with the object query q_j. For intuitive understanding, the attention mask in Fig. 2b displays 1 and 0 corresponding to 0 and $-\infty$ in (3), respectively.

3.2.2 Object Query Shuffle

Groups of object queries are assigned artificially. However, during the downstream object detection tasks, there are no explicit group assignment between object queries. Therefore, To simulate implicit group assignment between object queries, we randomly shuffle the permutation of all the object query embeddings during pre-training [3].

Fig. 2b illustrates the pre-training of multi-query patches with attention mask and object query shuffle. To improve the generalization, we randomly mask 10% query patches to zero during pre-training similarly to dropout [36]. In our experiments, two typical values are set to $N = 100$ and $M = 10$. Apart from such modifications, other training settings are the same as those described in Section 3.1.

4. Experiments

We pre-train the UP-DETR using ImageNet [10] and fine-tune the parameters on VOC [13] and COCO [28] for object detection, one-shot detection and panoptic segmentation. In all experiments, we adopt the UP-DETR model (41.3M parameters) with ResNet-50 backbone, 6 transformer encoder, 6 decoder layers of width 256 with 8 attention heads. Referring to the open source of DETR[4], we use *the same hyper-parameters* in the proposed UP-DETR

[3] In our further study, we find that the object query shuffle is not helpful. More details are included in the supplementary material.

[4] https://github.com/facebookresearch/detr

and our DETR re-implementation. We annotate R50 and R101 short for ResNet-50 and ResNet-101.

Pre-training setup. UP-DETR is pre-trained on the ImageNet training set without any labels. The CNN backbone (ResNet-50) is pre-trained with SwAV [7]. As the input image from ImageNet is relatively small, we resize it such that the shortest side is within $[320, 480]$ pixels while the longest side is at most 600 pixels. Given the image, we crop the query patches with random coordinate, height and width, which are resized to 128×128 pixels and transformed with the SimCLR-style [8] without horizontal flipping. AdamW [30] is used to optimize the UP-DETR, with the initial learning rate of 1×10^{-4} and the weight decay of 1×10^{-4}. We use a mini-batch size of 256 on 8 V100 GPUs for 60 epochs with the learning rate multiplied by 0.1 at 40 epochs.

Fine-tuning setup. The model is initialized with pre-training UP-DETR parameters and fine-tuned for all the parameters (including CNN) on VOC and COCO. We fine-tune the model with the initial learning rate 1×10^{-4} for transformers and 5×10^{-5} for CNN backbone, and the other settings are same as DETR [5] on 8 V100 GPUs. The model is fine-tuned with short/long schedule for 150/300 epochs and the learning rate is multiplied by 0.1 at 100/200 epochs, respectively.

4.1. PASCAL VOC Object Detection

Setup. The model is fine-tuned on VOC `trainval07+12` (~16.5k images) and evaluated on `test2007`. We report COCO-style metrics: AP, AP_{50} (default VOC metric) and AP_{75}. For a full comparison, we report the result of Faster R-CNN with the R50-C4 backbone [7], which performs much better than R50 [25]. DETR with R50-C4 significantly increases the computational cost than R50, so we fine-tune UP-DETR with R50 backbone.

Model/Epoch	AP	AP_{50}	AP_{75}
Faster R-CNN	56.1	**82.6**	**62.7**
DETR/150	49.9	74.5	53.1
UP-DETR/150	**56.1** (+6.2)	79.7 (+5.2)	60.6 (+7.5)
DETR/300	54.1	78.0	58.3
UP-DETR/300	**57.2** (+3.1)	80.1 (+2.1)	62.0 (+3.7)

Table 1: Object detection results trained on PASCAL VOC `trainval07+12` and evaluated on `test2007`. DETR and UP-DETR use R50 backbone and Faster R-CNN uses R50-C4 backbone. The values in the brackets are the gaps compared to DETR with the same training schedule.

Results. Table 1 shows the compared results of PASCAL VOC. We find that the DETR performs poorly in PASCAL VOC, which is much worse than Faster R-CNN by a large gap in all metrics. UP-DETR significantly boosts the performance of DETR for both short and long schedules: up to **+6.2 (+3.1)** AP, **+5.2 (+2.1)** AP_{50} and **+7.5 (+3.7)** AP_{75} for 150 (300) epochs, respectively. Moreover, UP-DETR (R50) achieves a comparable result to Faster R-CNN (R50-C4) with better AP. We find that both UP-DETR and DETR perform a little worse than Faster R-CNN in AP_{50} and AP_{75}. It may come from different ratios of feature maps (C4 for Faster R-CNN) and no NMS post-processing (NMS lowers AP but slightly improves AP_{50}).

Fig. 3a shows the AP (COCO style) learning curves on VOC. UP-DETR significantly speeds up the model convergence. After the learning rate reduced, UP-DETR significantly boosts the performance of DETR with a large AP improvement. Noting that UP-DETR obtains 56.1 AP after 150 epochs, however, its counterpart DETR (scratch transformers) only obtains 54.1 AP even after 300 epochs and does not catch up even training longer. It suggests that pre-training transformers is indispensable on insufficient training data (*i.e.* $\sim$ 16.5K images on VOC).

4.2. COCO Object Detection

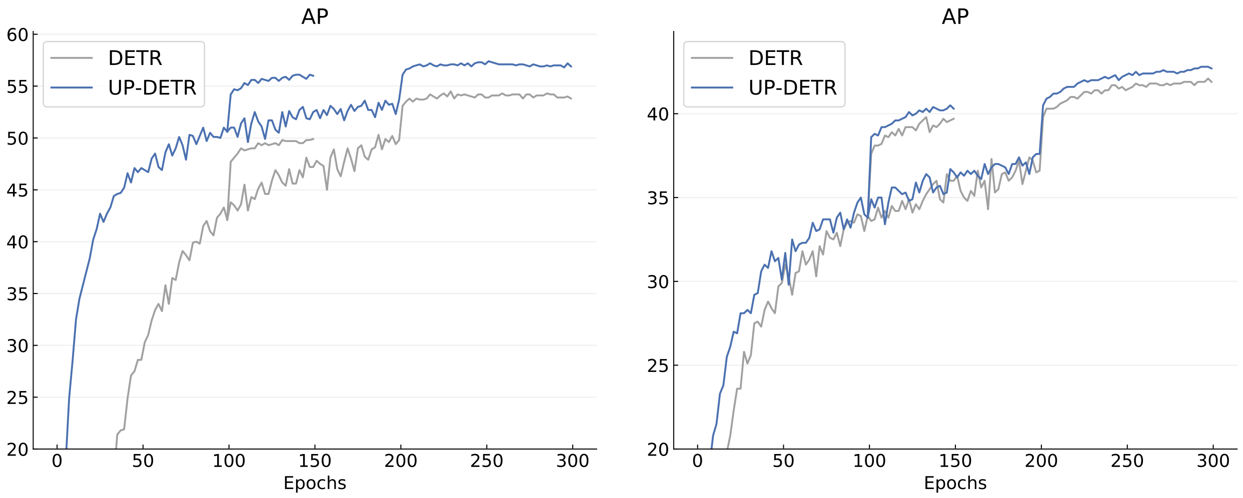
Setup. The model is fine-tuned on COCO `train2017` (~118k images) and evaluated on `val2017`. There are lots of small objects in COCO dataset, where DETR performs poorly [5]. Therefore, we report AP, AP_{50}, AP_{75}, AP_S, AP_M and AP_L for a comprehensive comparison. Moreover, we also report the results of highly optimized Faster R-CNN-FPN with short (3×) and long (9×) training schedules, which are known to improve the performance results [17].

Results. Table 2 shows the results on COCO with other methods. With 150 epoch schedule, UP-DETR outperforms DETR by 0.8 AP and achieves a comparable performance as compared with Faster R-CNN-FPN (3 × schedule). With 300 epoch schedule, UP-DETR obtains **42.8** AP on COCO, which is 0.7 AP better than DETR (SwAV CNN) and 0.8 AP better than Faster R-CNN-FPN (9 × schedule). Overall, UP-DETR comprehensively outperforms DETR in detection of small, medium and large objects with both short and long training schedules. Regrettably, UP-DETR is still slightly lagging behind Faster R-CNN in AP_S, because of the lacking of FPN-like architecture [26] and the high-cost attention operation.

Fig. 3b shows the AP learning curves on COCO. UP-DETR outperforms DETR for both 150 and 300 epoch schedules with faster convergence. The performance improvement is more noticeable before reducing the learning rate. After reducing the learning rate, UP-DETR still holds the lead of DETR by $\sim$ 0.7 AP improvement. It suggests that pre-training transformers is still indispensable even on sufficient training data (*i.e.* $\sim$ 118K images on COCO).

Model	Backbone	Epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster R-CNN † [26]	R101-FPN	-	36.2	59.1	39.0	18.2	39.0	48.2
Mask R-CNN † [18]	R101-FPN	-	38.2	60.3	41.7	20.1	41.1	50.2
Grid R-CNN † [31]	R101-FPN	-	41.5	60.9	44.5	23.3	44.9	53.1
Double-head R-CNN [40]	R101-FPN	-	41.9	62.4	45.9	23.9	45.2	55.8
RetinaNet † [27]	R101-FPN	-	39.1	59.1	42.3	21.8	42.7	50.2
FCOS † [38]	R101-FPN	-	41.5	60.7	45.0	24.4	44.8	51.6
DETR [5]	R50	500	42.0	62.4	44.2	20.5	45.8	61.1
Faster R-CNN	R50-FPN	3×	40.2	**61.0**	**43.8**	**24.2**	43.5	52.0
DETR (Supervised CNN)	R50	150	39.5	60.3	41.4	17.5	43.0	59.1
DETR (SwAV CNN) [7]	R50	150	39.7	60.3	41.7	18.5	43.8	57.5
UP-DETR	R50	150	**40.5** (+0.8)	60.8	42.6	19.0	**44.4**	**60.0**
Faster R-CNN	R50-FPN	9×	42.0	62.1	**45.5**	**26.6**	45.4	53.4
DETR (Supervised CNN)	R50	300	40.8	61.2	42.9	20.1	44.5	60.3
DETR (SwAV CNN) [7]	R50	300	42.1	**63.1**	44.5	19.7	46.3	60.9
UP-DETR	R50	300	**42.8** (+0.7)	63.0	45.3	20.8	**47.1**	**61.7**

Table 2: Object detection results trained on COCO train2017 and evaluated on val2017. Faster R-CNN, DETR and UP-DETR are performed under comparable settings. † for values evaluated on COCO test-dev, which are always slightly higher than val2017. The values in the brackets are the gaps compared to DETR.



(a) AP learning curves of VOC

(b) AP learning curves of COCO

Figure 3: AP (COCO style) learning curves with DETR and UP-DETR on VOC and COCO. Models are trained with the SwAV pre-training ResNet-50 for 150 and 300 epochs, and the learning rate is reduced at 100 and 200 epochs, respectively.

4.3. One-Shot Detection

Given a query image patch whose class label is not included in the training data, one-shot detection aims to detect all instances with the same class in a target image. One-shot detection is a promising research direction that can detect unseen instances. With feeding query patches to the decoder, UP-DETR is naturally compatible to one-shot detection task. Therefore, one-shot detection can also be treated as a downstream fine-tuning task of UP-DETR.

Following the same one-shot detection setting as [20], we crop the query image patch as the query patch to the DETR decoder. we train DETR and UP-DETR on VOC 2007train val and 2012train val sets with 300 epochs then evaluate on VOC 2007test set. Table 3 shows the comparison to the state-of-the-art one-shot detection methods. Compared with DETR, UP-DETR significantly boosts the performance of DETR on both seen (**+22.8** AP^{50} gain) and unseen (**+15.8** AP^{50} gain) classes. Moreover, we show that UP-DETR outperforms all methods in both seen (**+7.9** AP^{50} gain) and unseen (**+4.0** AP^{50} gain)

Model	seen class																unseen class				
	plant	sofa	tv	car	bottle	boat	chair	person	bus	train	horse	bike	dog	bird	mbike	table	**AP**^{50}	cow	sheep	cat	aero	**AP**^{50}
SiamFC [2]	3.2	22.8	5.0	16.7	0.5	8.1	1.2	4.2	22.2	22.6	35.4	14.2	25.8	11.7	19.7	27.8	15.1	6.8	2.28	31.6	12.4	13.3
SiamRPN [23]	1.9	15.7	4.5	12.8	1.0	1.1	6.1	8.7	7.9	6.9	17.4	17.8	20.5	7.2	18.5	5.1	9.6	15.9	15.7	21.7	3.5	14.2
CompNet [47]	28.4	41.5	65.0	66.4	37.1	49.8	16.2	31.7	69.7	73.1	75.6	71.6	61.4	52.3	63.4	39.8	52.7	75.3	60.0	47.9	25.3	52.1
CoAE [20]	30.0	54.9	64.1	66.7	40.1	54.1	14.7	60.9	77.5	78.3	77.9	73.2	80.5	70.8	**72.4**	46.2	60.1	**83.9**	67.1	75.6	46.2	68.2
Li *et al.* [24]	33.7	58.2	67.5	72.7	40.8	48.2	20.1	55.4	**78.2**	79.0	76.2	74.6	**81.3**	71.6	72.0	48.8	61.1	74.3	68.5	**81.0**	52.4	69.1
DETR	11.4	42.2	44.1	63.4	14.9	40.6	20.6	63.7	62.7	71.5	59.6	52.7	60.6	53.6	54.9	22.1	46.2	62.7	55.2	65.4	45.9	57.3
UP-DETR	**46.7**	**61.2**	**75.7**	**81.5**	**54.8**	**57.0**	**44.5**	**80.7**	74.5	**86.8**	**79.1**	**80.3**	80.6	**72.0**	70.9	**57.8**	**69.0**	80.9	**71.0**	80.4	**59.9**	**73.1**

Table 3: One-shot detection results on VOC `2007test` set.

Model	PQ	SQ	RQ	PQth	SQth	RQth	PQst	SQst	RQst	AP^{seg}
PanopticFPN++ [21]	42.4	79.3	51.6	49.2	82.4	58.8	32.3	74.8	40.6	**37.7**
UPSNet [42]	42.5	78.0	52.5	48.6	79.4	59.6	33.4	75.9	41.7	34.3
UPSNet-M [42]	43.0	79.1	52.8	48.9	79.7	59.7	34.1	78.2	42.3	34.3
DETR [5]	44.3	80.0	54.5	49.2	80.6	60.3	**37.0**	**79.1**	**45.9**	32.9
UP-DETR	**44.5**	**80.3**	**54.7**	**49.6**	**80.7**	**60.7**	36.9	78.9	45.8	34.0

Table 4: Panoptic segmentation results on the COCO `val` dataset with the same ResNet-50 backbone. The PanopticFPN++, UPSNet and DETR results are re-implemented by Carion *et al.* [5].

Case	Frozen CNN	Feature Reconstruction	AP$_{50}$
DETR	scratch transformers		74.5
(a)			74.0
(b)	✓		**78.7**
(c)		✓	62.0
(d)	✓	✓	**78.7**

Table 5: Ablation study on frozen CNN and feature reconstruction for pre-training models with AP$_{50}$. The experiments are fine-tuned on PASCAL VOC with 150 epochs.

classes of one-hot detection. It further verifies the effectiveness of our pre-training pretext task.

4.4. Panoptic Segmentation

Panoptic segmentation [21] is a natural extension to DETR by adding a mask head on the top of the decoder outputs. Following the same panoptic segmentation training schema as DETR [5], we fine-tune UP-DETR for box only annotations with 300 epochs. Then, we freeze all the weights of DETR and train the mask head for 25 epochs.

Table 4 shows the comparison to state-of-the-art methods on panoptic segmentation with the ResNet-50 backbone. As seen, UP-DETR outperforms DETR[5] with **+0.2** PQ, **+0.4** PQth and **+1.1** AP^{seg}.

4.5. Ablations

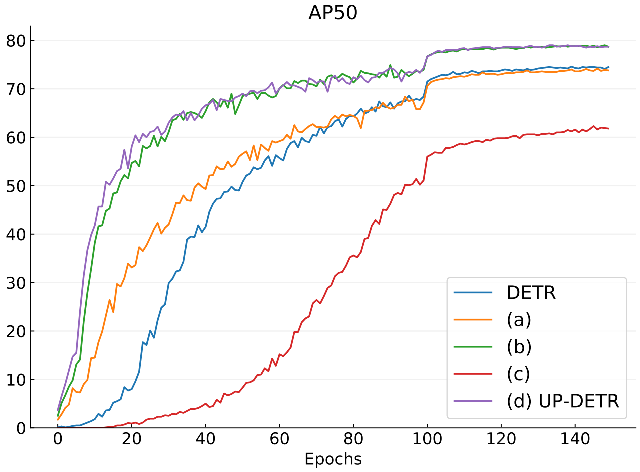


Figure 4: Learning curves of VOC (AP$_{50}$) with four different pre-training UP-DETR models and DETR. The models trained with 150 epochs corresponds to the models in Table 5 one-to-one.

For ablation experiments, we pre-train UP-DETR for 15 epochs with the learning rate multiplied by 0.1 at the 10-th epoch on ImageNet. We fine-tune models on VOC object detection following the setup in Section 4.1 with 150 epochs[6].

4.5.1 Frozen CNN and Feature Reconstruction

To illustrate the importance of patch feature reconstruction and frozen CNN backbone of UP-DETR, we pre-train four different UP-DETR models with different combinations of whether freezing CNN and whether adding feature reconstruction.

[5] With a bug fixed in `github.com/facebookresearch/detr/issues/247`, the DETR baseline is better than paper report.

[6] More ablations are included in the supplementary material.

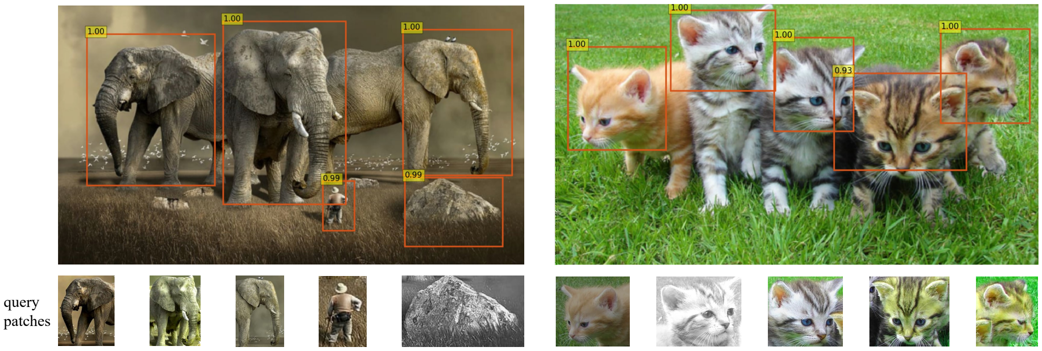


Figure 5: The unsupervised localization of patch queries with UP-DETR. The first line is the original image with predicted bounding boxes. The second line is query patches cropped from the original image with data augmentation. The value in the upper left corner of the bounding box is the model confidence.

Table 5 shows AP and AP_{50} of four different pre-training models and DETR on VOC with 150 epochs. As shown in Table 5, not all pre-trained models are better than DETR, and pre-training models (b) and (d) perform better than the others. More importantly, without frozen CNN, pre-training models (a) and (c) even perform worse than DETR. It confirms that freezing pre-trained CNN is essential to pre-train transformers. In addition, it further confirms the pretext (random query patch detection) may weaken the feature discrimination without the freezing pre-training CNN weights.

Fig. 4 plots the AP_{50} learning curves of four different pre-training models and DETR, where the models in Fig. 4 correspond to the models in Table 5 one-to-one. As shown in Fig. 4, model (d) UP-DETR achieves faster convergence at the early training stage with feature reconstruction. The experiments suggest that random query patch detection is complementary to the contrastive learning for a better visual representation. The former is designed for the spatial localization with position embeddings, and the latter is designed for instance or cluster classification.

It is worth noting that UP-DETR with frozen CNN and feature reconstruction heavily relies on a pre-trained CNN model, *e.g*. SwAV. Therefore, we believe that it is a promising direction for further investigating UP-DETR with random query patch detection and contrastive learning together to pre-train the whole DETR model from scratch.

4.6. Visualization

To further illustrate the ability of the pre-training model, we visualize the unsupervised localization results of given patch queries. Specifically, for the given image, we manually crop several object patches and apply the data augmentation to them. Then, we feed these patches as queries to the model. Finally, we visualize the model output with bounding boxes, whose classification confidence is greater than 0.9. This procedure can be treated as *unsupervised one-shot detection* or deep learning based *template matching*.

As shown in Fig. 5, pre-trained with random query patch detection, UP-DETR successfully learns to locate the bounding box of given query patches and suppress the duplicated bounding boxes [7]. It suggests that UP-DETR with random query patch detection is effective to learns the ability of object localization.

5. Conclusion

We present a novel pretext task called random query patch detection to pre-train the transformers in DETR. With unsupervised pre-training, UP-DETR significantly outperforms DETR on object detection, one-shot detection and panoptic segmentation. We find that, even on the COCO with sufficient training data, UP-DETR still performs better than DETR.

From the perspective of unsupervised pre-training models, pre-training CNN backbone and pre-training transformers are separated now. Recent studies of unsupervised pre-training mainly focus on feature discrimination with contrastive learning instead of specialized modules for spatial localization. However, for UP-DETR pre-training, the pretext task is mainly designed for patch localization by positional encodings and learn-able object queries. We hope that an advanced method can integrate CNN and transformers pre-training into a unified end-to-end framework and apply our pre-training tasks to more detection related frameworks.

Acknowledgement

This work was supported by the Guangdong Natural Science Foundation under Grant 2019A1515012152.

[7] Base picture credit: https://www.piqsels.com/en/public-domain-photo-jrkkq, https://www.piqsels.com/en/public-domain-photo-smdfn.

References

[1] YM Asano, C Rupprecht, and A Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *International Conference on Learning Representations*, 2019. 2

[2] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *European conference on computer vision*, pages 850–865. Springer, 2016. 7

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into high quality object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6154–6162, 2018. 2

[4] Yue Cao, Zhenda Xie, Bin Liu, Yutong Lin, Zheng Zhang, and Han Hu. Parametric instance classification for unsupervised visual feature learning. *Advances in Neural Information Processing Systems*, 33, 2020. 2

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *arXiv preprint arXiv:2005.12872*, 2020. 1, 2, 3, 5, 6, 7

[6] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018. 2

[7] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33, 2020. 1, 2, 5, 6

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 2, 5

[9] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 1

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee, 2009. 4

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1, 2

[12] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, pages 13063–13075, 2019. 4

[13] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 1, 2, 4

[14] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33, 2020. 2

[15] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006. 2

[16] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 1, 2

[17] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4918–4927, 2019. 5

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969, 2017. 6

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1

[20] Ting-I Hsieh, Yi-Chen Lo, Hwann-Tzong Chen, and Tyng-Luh Liu. One-shot object detection with co-attention and co-excitation. *arXiv preprint arXiv:1911.12529*, 2019. 6, 7

[21] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019. 7

[22] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 734–750, 2018. 2

[23] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8971–8980, 2018. 7

[24] Xiang Li, Lin Zhang, Yau Pun Chen, Yu-Wing Tai, and Chi-Keung Tang. One-shot object detection without fine-tuning. *arXiv preprint arXiv:2005.03819*, 2020. 7

[25] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2359–2367, 2017. 5

[26] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017. 5, 6

[27] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988, 2017. 2, 6

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence

Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014. 1, 2, 4

[29] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, pages 21–37. Springer, 2016. 2

[30] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018. 5

[31] Xin Lu, Buyu Li, Yuxin Yue, Quanquan Li, and Junjie Yan. Grid r-cnn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7363–7372, 2019. 6

[32] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018. 1, 2

[33] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019. 1, 2

[34] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015. 1, 2

[35] Guanglu Song, Yu Liu, and Xiaogang Wang. Revisiting the sibling head in object detector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11563–11572, 2020. 2, 4

[36] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 4

[37] Russell Stewart, Mykhaylo Andriluka, and Andrew Y Ng. End-to-end people detection in crowded scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2325–2333, 2016. 2, 3

[38] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9627–9636, 2019. 2, 6

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. 1, 2

[40] Yue Wu, Yinpeng Chen, Lu Yuan, Zicheng Liu, Lijuan Wang, Hongzhi Li, and Yun Fu. Rethinking classification and localization for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10186–10195, 2020. 2, 4, 6

[41] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018. 2

[42] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified panoptic segmentation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8818–8826, 2019. 7

[43] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, pages 5753–5763, 2019. 2

[44] Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6210–6219, 2019. 2

[45] Haichao Zhang and Jianyu Wang. Towards adversarially robust object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 421–430, 2019. 2, 4

[46] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9759–9768, 2020. 1, 2

[47] Tengfei Zhang, Yue Zhang, Xian Sun, Hao Sun, Menglong Yan, Xue Yang, and Kun Fu. Comparison network for one-shot conditional object detection. *arXiv e-prints*, pages arXiv–1904, 2019. 7

[48] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 2